

An adaptive pathline-based particle tracking algorithm for the Eulerian–Lagrangian method

Jacob Bensabat^{a,*}, Quanlin Zhou^b, Jacob Bear^b

^a *Environmental and Water Resources Engineering Inc., P.O. Box 6770, Haifa 31067, Israel*

^b *Faculty of Civil Engineering, Technion-Israel Institute of Technology, Haifa 32000, Israel*

Received 30 July 1998; received in revised form 18 March 1999; accepted 25 April 1999

Abstract

The accuracy in determining the Lagrangian concentration in the Eulerian–Lagrangian method depends on both the particle tracking algorithm and concentration interpolation. In most existing particle tracking algorithms, accurate tracking cannot be achieved, particularly in a complicated unsteady flow. Also mechanisms for error estimation and accuracy control are not available. A new algorithm is developed by refining the process of particle tracking, making use of a series of available travel time increments (ATTIs). The ATTIs are selected on the basis of tracking accuracy and efficiency. These are controlled by practical indices related to the rate of particle velocity variation (magnitude and direction). The particle velocity is determined by bilinear interpolation in space and time. The proposed algorithm combines an inter-element refinement, consistent with the piecewise interpolation of velocity, and an in-element pathline-based refinement. No mesh refinement is needed. Numerical simulations have been used to demonstrate the accuracy of the proposed algorithm. A comparison between the results obtained by using the proposed algorithm and by some existing techniques shows that the new algorithm can provide an accurate and efficient particle tracking, especially in a complicated unsteady heterogeneous flow. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Eulerian–Lagrangian method; Particle tracking; Pathline; Mesh refinement; Adaptation; Numerical simulation

1. Introduction

The Eulerian–Lagrangian method (ELM – a list of all acronyms is given in Appendix A) is a potentially powerful framework for the solution of advection–dispersion equations. This method is based on the decomposition of the concentration field into two components: advection and dispersion. The advection-based concentration (Lagrangian concentration) is computed by using a particle tracking technique, while the dispersion problem is solved by a finite element (FEM), or a finite difference (FDM) method. The accuracy of the Lagrangian concentration depends on the employed particle tracking and concentration interpolation techniques. The latter can be improved by combining both forward and backward particle tracking methods in regions where sharp concentration fronts or concentration peaks/valleys occur [11,16,17,19].

A number of particle tracking techniques have been developed and reported in the literature [2,4,5,7,8,10–

14,16–19]. These techniques can be classified as analytical, semi-analytical, and numerical integration methods.

1.1. Analytical method

In this method [2], the tracking of particles is accomplished by solving for the stream function and then computing streamlines and travel times of contaminants. The method gives an exact solution, and no computational error is involved. However, most analytical solutions for pathlines are limited to one- and two-dimensional steady state flows, with simple geometry and hydrogeological conditions.

1.2. Semi-analytical method

In this approach [5–8,10,13,14], the particle path and travel time within a cell are obtained by analytical integration. Linear interpolation is employed in order to compute the velocity field by the finite difference or the mixed finite element method. A constant velocity at each cell face is used to represent the local velocity field in a

* Corresponding author. Tel.: +972-4-838-1515; fax: +972-4-838-7621
E-mail address: jbensabat@ewre.com (J. Bensabat)

cell (element). The particle velocity in the x - (y -, or z -) direction depends only on the x (y , or z) coordinate of the particle's position. This linear interpolation yields a discontinuous velocity field. Any component of the velocity vector undergoes an abrupt change as a particle crosses a cell boundary, which is parallel to the component direction. This method is not applicable to bilinearly and trilinearly interpolated velocity fields on the basis of face velocity. Moreover, it cannot be used in a nodal velocity field in the finite element method, since the velocity at an arbitrary point depends on its location.

1.3. Numerical integration method

In a non-uniform velocity field, the particle velocity is nonlinear with respect to travel time. In general, this relationship cannot be analytically derived. Euler (or single-velocity), Euler predictor–corrector (or average-velocity), and Runge–Kutta methods have been widely used for the numerical integration in the particle tracking process [4,7,8,12,18,19]. In the Euler method, the particle velocity at its starting point is used to approximate the tracking velocity within a tracking step. It is very efficient when applied to a smooth flow field. However, it may result in large errors when the velocity undergoes significant variations in both magnitude and direction. In the Euler predictor–corrector method, the particle velocity at the starting point is used for predicting the particle's position at a trial end point. The average-velocity, obtained from the particle velocities at both the starting and the trial end points, is used to correct the predicted value. This iterative correction procedure continues until convergence is achieved. In the explicit fourth-order Runge–Kutta method, the particle velocity is evaluated four times for each tracking step: once at the starting point, twice at two trial mid-points, and once at a trial end point. A weighted tracking velocity, based on values evaluated at these four points, is used for tracking the particle to its new position.

The accuracy of the particle tracking phase is crucial for the solution of the advection–dispersion problems, as the ELM does not inherently conserve mass, either locally or globally. Oliveira and Baptista [12] systematically investigated the effect of inaccurate tracking on the numerical properties of the ELM. They showed that particle tracking errors not only affect mass, but can also introduce significant phase errors and numerical dispersion. They may lead to instability of an otherwise stable ELM. However, in most existing models, no systematic effort has been made for error estimation and accuracy control in particle tracking. Tracking errors depend on the complexity of the local velocity field and on the size of a tracking time step. In a uniform velocity field, an exact result can be obtained by any numerical tracking method. However, in the vicinity of a divergent

flow around a source, or a convergent one around a sink, the local velocity field undergoes significant variations in both magnitude and direction. The tracking process in a transport simulation step should be refined into a large number of tracking steps. To achieve this goal, a local mesh refinement has been suggested by Cheng et al. [4]. However, this mesh refinement does not introduce any new information on the velocity field, since the flow field is not recomputed at refined nodes.

In this paper, an adaptive *pathline-based* particle tracking algorithm is presented. The particle *tracking process* in a transport simulation time step (TSS) is refined, along a particle path, into a number of *tracking steps*, by splitting the TSS into the same number of travel time increments (TTIs). The proposed algorithm is consistent with the piecewise interpolation of a local velocity field by splitting the tracking process at element boundaries (*inter-element refinement*). The inter-element and the in-element path refinements are combined, and no mesh refinement is needed. Two practical indices of particle velocity (the rate of variation in its magnitude and direction) are provided to control tracking errors. Adjacent tracking times (ATTIs) are adapted on the basis of the accuracy and efficiency of the tracking process. The particle velocity at an arbitrary point is determined by a bilinear interpolation in both space and time. The implementation of the proposed algorithm is discussed in detail in Sections 2 and 3. This algorithm is then compared with existing ones for 2-D steady and unsteady complicated flows (Section 4).

2. Theory

We start with a known velocity field, which may have been calculated either analytically, or by any of the existing numerical procedures [1–3,5,6,8,15,20]. The proposed algorithm can be applied in either the finite difference or the finite element framework. It can be used for both forward and backward particle tracking. In what follows, we shall demonstrate this algorithm for forward particle tracking, with the average-velocity method, within the framework of the finite element technique.

2.1. Particle velocity

Assume that we have obtained a nodal velocity field by solving a flow problem, either analytically or numerically. In a steady flow, the particle velocity within a visited element is determined by the interpolation scheme

$$V^P(\xi^P(t)) = \sum_{J=1}^{M^e} N_J^e(\xi^P(t)) V_J, \quad (1)$$

where $\xi^P(t)$ is the particle’s position vector at time t ($T_n \leq t \leq T_{n+1}$), T_n and T_{n+1} are the starting and end times of the n th TSS, with $\Delta T_n = T_{n+1} - T_n$, $\mathbf{V}^P(\xi^P(t))$ is the particle velocity vector at the position $\xi^P(t)$, \mathbf{V}_J is the computed local nodal velocity vector at the J th node in an element e , N_J^e is the local basis function, and M^e is the number of element nodes. Here, the particle velocity depends only on the particle’s position.

In an unsteady flow, the particle velocity at time t is determined by using bilinear interpolation in both space and time

$$\mathbf{V}^P(\xi^P(t), t) = \sum_{J=1}^{M^e} N_J^e(\xi^P(t)) \left\{ \frac{T_{n+1} - t}{\Delta T_n} \mathbf{V}_J(T_n) + \frac{t - T_n}{\Delta T_n} \mathbf{V}_J(T_{n+1}) \right\}, \tag{2}$$

where $\mathbf{V}_J(T_n)$ and $\mathbf{V}_J(T_{n+1})$ are the local nodal velocity vectors at the J th node at the beginning and end times of the single TSS, respectively. Here the particle velocity depends on both the particle’s position $\xi^P(t)$ and time t . For simplicity, we use the notation

$$\mathbf{V}^P(t) = \begin{cases} \mathbf{V}^P(\xi^P(t)) & \text{for a steady flow,} \\ \mathbf{V}^P(\xi^P(t), t) & \text{for an unsteady flow.} \end{cases}$$

In a non-uniform velocity field, the relationship between a particle’s displacement and its travel time is nonlinear, and, thus, the particle velocity is also nonlinear with respect to the travel time.

2.2. Particle tracking

The particle velocity can be expressed as the time derivative of the particle’s displacement

$$\frac{d\xi^P(t)}{dt} = \mathbf{V}^P(t). \tag{3}$$

Eq. (3) can be integrated for the n th TSS

$$\xi^P(T_{n+1}) = \xi^P(T_n) + \int_{T_n}^{T_{n+1}} \mathbf{V}^P(t) dt, \tag{4}$$

where $\xi^P(T_n)$ and $\xi^P(T_{n+1})$ are the particle’s position vectors at its starting and end points, at times T_n and T_{n+1} , respectively.

Let us split the single TSS into N_t non-uniform TTIs (as described in the next section). Eq. (4) is then rewritten as

$$\xi^P(T_{n+1}) = \xi^P(T_n) + \sum_{i=1}^{N_t} \int_{t_i}^{t_{i+1}} \mathbf{V}^P(t) dt, \tag{5}$$

where t_i and t_{i+1} are the starting and end times of the i th tracking step, $t_i = T_n + \sum_{k=1}^{i-1} \Delta t_k$ ($i = 1, N_t$), and Δt_k is the length of the k th TTI.

Let us define a *mean tracking velocity*, $\tilde{\mathbf{V}}_i^P$, in a tracking step (t_i, t_{i+1}) , by

$$\tilde{\mathbf{V}}_i^P = \frac{1}{\Delta t_i} \int_{t_i}^{t_{i+1}} \mathbf{V}^P(t) dt. \tag{6}$$

Then, the forward particle tracking process during the n th TSS takes the form

$$\xi^P(T_{n+1}) = \xi^P(T_n) + \sum_{i=1}^{N_t} \tilde{\mathbf{V}}_i^P \Delta t_i, \tag{7}$$

and the particle tracking in the i th tracking step is expressed by

$$\xi^P(t_{i+1}) = \xi^P(t_i) + \tilde{\mathbf{V}}_i^P \Delta t_i. \tag{8}$$

2.3. Error estimation and accuracy control

We linearize the nonlinear relationship between the particle velocity and the travel time, and approximate the mean tracking velocity $\tilde{\mathbf{V}}_i^P$, using a simple average velocity $\bar{\mathbf{V}}_i^P$ in the Euler predictor–corrector method

$$\begin{aligned} \tilde{\mathbf{V}}_i^P &\approx \bar{\mathbf{V}}_i^P = \frac{1}{2} (\mathbf{V}^P(t_i) + \mathbf{V}^P(t_{i+1})) \\ &= \mathbf{V}^P(t_i) + \frac{1}{2} (\mathbf{V}^P(t_{i+1}) - \mathbf{V}^P(t_i)). \end{aligned} \tag{9}$$

This approximation may produce a large error in determining the particle’s position at the end point in a tracking step. The truncated error of this approximation may be estimated by Taylor’s expansion and Richardson method [9]

$$\varepsilon_T = -\frac{1}{12} \frac{d^2 \mathbf{V}^P(\tau)}{d\tau^2} \Delta t_i^3, \tag{10}$$

$$\varepsilon_R = \frac{\xi^{P^*}(t_{i+1}) - \xi^P(t_{i+1})}{3}, \tag{11}$$

where $\tau \in [t_i, t_{i+1}]$, and $\xi^{P^*}(t_{i+1})$ is the solution by two successive tracking steps, with a travel time increment of $\Delta t_i/2$. It is found that the second-order derivative of the particle velocity is proportional to the particle velocity, and is related to the heterogeneity of the local velocity field. However, the evaluation of the second derivative of the particle velocity vector is rather complicated for multi-dimensional unsteady flow, and the evaluation of ε_T requires some algebra. The evaluation of ε_R requires two additional tracking steps. Therefore, these two methods require additional computational work, which may affect the algorithm efficiency. Thus, (10) and (11) cannot be efficiently used for the adaptation of the size of the tracking time step, which is an operation that is performed a large number of times, particularly in a 3-D procedure.

In a real non-uniform velocity field, and a smooth local velocity field, particles may be accurately tracked by the inter-element refinement. Only in a local velocity field, with significant variations in velocity magnitude and/or direction, has the particle tracking process to be further refined along the particle’s path. In the proposed

algorithm, we first analyze the potential tracking errors, and then provide two simple and practical indices for controlling such errors.

In a steady flow, the particle's position at the end of a tracking step can be exactly determined by the average velocity, \bar{V}_i^P , and the particle path can be exact. However, the travel time between the starting and end points cannot be estimated accurately.

In a one-dimensional linearly interpolated flow (Fig. 1(a)), the tracking velocity can be analytically determined [8,10,13,14] by

$$\bar{V}_i^P = \begin{cases} \Delta V_i^P \{ \ln(1 + r_i) \}^{-1}, & r_i \neq 0, \\ \bar{V}_i^P = V^P(t_i), & r_i = 0, \end{cases} \quad (12)$$

where $r_i = \Delta V_i^P / V^P(t_i)$ is the rate of variation in the particle velocity, and $\Delta V_i^P = V^P(t_{i+1}) - V^P(t_i)$. If a linear approximation (\bar{V}_i^P) is used, the travel time for a given displacement along the particle path is always underestimated, or the particle passes beyond the exact end point for a given travel time. The larger the rate of variation in the particle velocity is, the larger is the error in the approximation of \bar{V}_i^P by \bar{V}_i^P . For example, when $r_i = 1$, \bar{V}_i^P produces a 4% relative error in the tracking velocity, and when $r_i = 10$ leads to a 44% relative error with respect to the linear approximation.

In multi-dimensional flow, without directional variation in the particle velocity, (12) can be obtained by a coordinate transformation. The error resulting from the linear approximation is related to the rate of variation of the particle velocity.

In a multi-dimensional flow, with variations in both magnitude and direction of particle velocity, the travel time for a given displacement may be underestimated or overestimated, depending on two kinds of errors: (1) the difference between the particle velocity on the tracked path and that on the exact path; and (2) the difference between \bar{V}_i^P and \bar{V}_i^P along the tracked path. At a given time, both errors may compensate each other, or they

may combine, thus increasing the error in the evaluation of the travel time. For example, in the case shown in Fig. 1(b), they balance each other, producing an over-estimated travel time.

In unsteady flow, a distortion of the tracked path may occur if the travel time is not estimated accurately (Fig. 1(c)). The particle velocity depends on the spatial and the temporal interpolations, calculated on the basis of its position and travel time. It may temporally change its direction at a point, thus producing a difference between the tracked and the exact particle paths.

Therefore, a potential error may result from the nonlinearity in particle velocity with respect to travel time, and from the approximation of \bar{V}_i^P by \bar{V}_i^P . This error depends on the rate of variation in magnitude, r_i , and direction, θ_i , of the particle velocity, where $r_i = \max_j(r_{ij})$, (13)

with

$$r_{ij} = \frac{|V_j^P(t_{i+1}) - V_j^P(t_i)|}{\min(|V_j^P(t_i)|, |V_j^P(t_{i+1})|)},$$

where $V_j^P(t_i)$ and $V_j^P(t_{i+1})$ are the j th components of the particle's velocities $V^P(t_i)$ and $V^P(t_{i+1})$, ($j = x, y$ in two dimensions, and $j = x, y, z$ in three dimensions). The variation in the direction of particle velocity is defined as

$$\theta_i = \arccos \left(\frac{V^P(t_i) \cdot V^P(t_{i+1})}{|V^P(t_i)| |V^P(t_{i+1})|} \right). \quad (14)$$

Note that the same error will result from $|\Delta V_i^P|$, whether the particle velocity increases or decreases.

Since an exact particle tracking can be achieved for a uniform flow field, the tracking error depends on r_i and θ_i . In what follows, we assume that the potential tracking error depends linearly on the rate of variation in magnitude and direction of the particle velocity. By controlling r_i and θ_i within predefined ranges, i.e., ensuring that

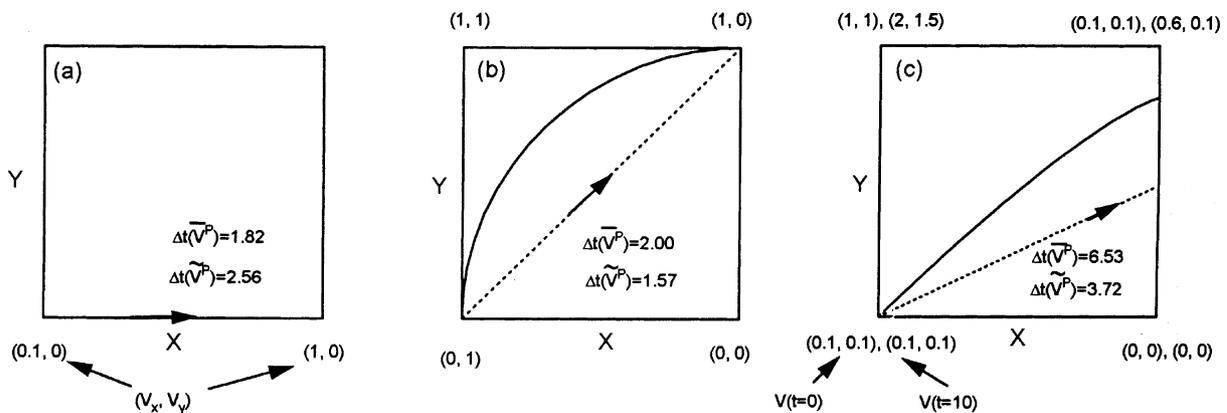


Fig. 1. Illustration of tracking errors in (a) 1-D steady flow, (b) 2-D steady flow, and (c) 2-D unsteady flow (the solid line indicates the exact path and the dotted line indicates the tracked path).

$$(a) r_i < R, \quad \text{and} \quad (b) \theta_i < \Theta, \quad (15)$$

we can use \bar{V}_i^P directly and obtain a high tracking accuracy. Here, R and Θ are the predefined bounds on r_i and θ_i , respectively. Controlling r_i and θ_i is equivalent to choosing a TTI such that the linear approximation of the tracking velocity produces accurate tracking in each tracking step. Note that R and Θ differ for different numerical integration methods (Euler, Euler predictor–corrector, and Runge–Kutta methods).

3. Implementation of the adaptive pathline-based particle tracking

At the beginning of the entire particle tracking process in the n th TSS, we initialize all particle's attributes to their values at the starting point. These particle's attributes include:

- starting time, t_i ;
- particle position at the start time, $\xi^P(t_i)$;
- particle velocity, $V^P(t_i)$;
- the element that includes the particle, P_{el} ;
- the geometric status, P_{st} , of the particle in the P_{el} -element (e.g., whether the particle is located at a global node, at an edge, (on a face in three dimensions), inside P_{el} -element, or on a boundary face of the considered domain);
- the set of nodes, P_{nd} , of P_{el} -element pointed by P_{st} (for example, in a 2-D quadrilateral element, the possible status of a particle are: (1) at a node, with one global node in P_{nd} ; (2) at an edge, with 2 global nodes in P_{nd} ; and (3) inside the element, with empty P_{nd} (shown in Fig. 4)).
- the ATTI, the size of the available travel time for a tracking step (δt_i).

The adaptive pathline-based particle tracking algorithm consists of three parts: (1) tracking action, (2) refinement and adaptation, and (3) updating the particle's attributes. This algorithm is illustrated in the flow chart of Fig. 2.

3.1. Tracking action

After an ATTI in a tracking step is determined, either initially or in the adaptation step, the tracking action is used to determine the new values of the particle's attributes at the trial end point. The tracking action is illustrated in the flow chart of Fig. 3.

Step 1: Determine eligible elements and inter-element boundaries, which may be visited by a particle.

We define an *eligible element* as an element that may be visited by a particle. An *eligible inter-element boundary* in an eligible element is one that the particle may

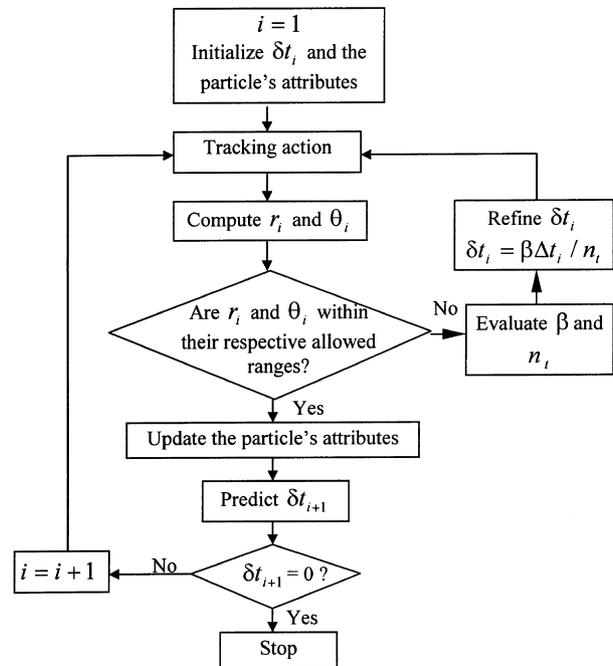


Fig. 2. Flow chart of the adaptive pathline-based particle tracking algorithm (PPT).

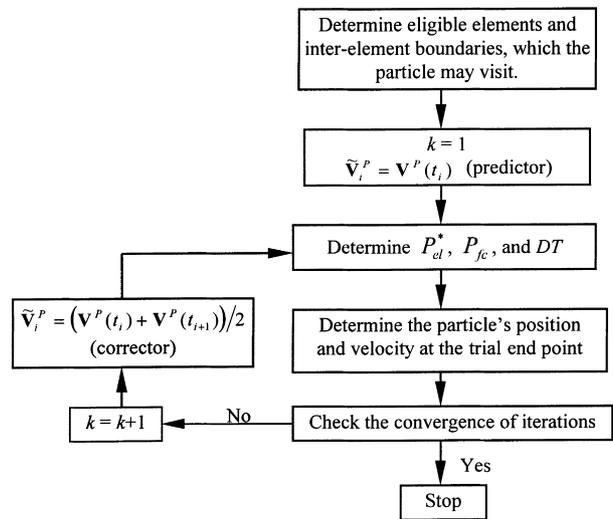


Fig. 3. Flow chart of a tracking action with the average-velocity tracking method.

cross. All elements connected with the particle are eligible. These elements are determined from the particle status P_{st} , and its connected global node set, P_{nd} . In each eligible element, only the inter-element boundaries, on which the particle is not located, are *eligible*; the other boundaries which are connected to the particle are *non-eligible*. It should be noted that a particle may end up at an eligible or a non-eligible inter-element boundary. The number of eligible elements and of eligible inter-element boundaries may differ for different values of P_{st} . For

example, when a particle is located on an edge in a quadrilateral element, which is not on a boundary, there are two eligible elements and three eligible edges for each eligible element (shown in Fig. 4). This geometrical analysis of the particle is based only on the particle's attributes, which are related to the mesh. The use of these concepts of eligible elements and inter-element boundaries facilitates the tracking of a particle in any kind of mesh structure.

Step 2: Determine the element and inter-element boundary, P_{el}^ and P_{fc} , through which a particle passes, and the travel time, DT , required for the particle to reach the P_{fc} -boundary.*

For a given tracking velocity, the method described below is used to determine the P_{el}^* -element, the P_{fc} -face, and DT .

In two dimensions, for the line J_1J_2 on an eligible edge in an eligible element (J_1 and J_2 are global nodes), the intersection point, Q , with the particle path is determined by the line equation:

$$x_Q = x_{J_1} + b(x_{J_2} - x_{J_1}), \quad y_Q = y_{J_1} + b(y_{J_2} - y_{J_1}), \quad (16)$$

and the path equation

$$x_Q = \xi_x^P(t_i) + d\tilde{V}_{ix}^P, \quad y_Q = \xi_y^P(t_i) + d\tilde{V}_{iy}^P, \quad (17)$$

where (x_{J_1}, y_{J_1}) and (x_{J_2}, y_{J_2}) are the coordinates of J_1 and J_2 , respectively, $\xi_x^P(t_i)$ and $\xi_y^P(t_i)$ are the components of the starting position of the particle, \tilde{V}_{ix}^P and \tilde{V}_{iy}^P are the components of the tracking velocity \tilde{V}_i^P , and b and d are two parameters. These two parameters are given by

$$b = \frac{(\xi_x^P(t_i) - x_{J_1}) + (\xi_y^P(t_i) - y_{J_1}) + d(\tilde{V}_{ix}^P + \tilde{V}_{iy}^P)}{(x_{J_2} - x_{J_1}) + (y_{J_2} - y_{J_1})} \quad (18)$$

and

$$d = \frac{(x_{J_2} - x_{J_1})(\xi_y^P(t_i) - y_{J_1}) - (y_{J_2} - y_{J_1})(\xi_x^P(t_i) - x_{J_1})}{\tilde{V}_{ix}^P(y_{J_2} - y_{J_1}) - \tilde{V}_{iy}^P(x_{J_2} - x_{J_1})}. \quad (19)$$

Note that when the denominators in (18) and (19) are zero, we use corresponding equations for one component (x , or y). We evaluate these two parameters for each eligible edge. The parameter b is used to check whether the particle path intercepts the considered eligible edge, while d is used to check whether the particle moves to or away from the edge. If $d < 0$, the particle will move away from the line J_1J_2 on the eligible edge (see Fig. 5). If $b < 0$ or $b > 1$, the particle cannot intercept the edge J_1J_2 . For an eligible edge, if

$$d > 0 \text{ and } 0 \leq b \leq 1, \quad (20)$$

the particle will reach the edge J_1J_2 at the exit point Q if the ATTI $\delta t_i \geq d$. In this case, d is the minimum among all positive values of d for all eligible edges, and it is equal to the travel time from the starting point S to Q ($DT = d$). The element and edge that satisfy (20) are the required ones for the particle to cross, the P_{el}^* -element and the P_{fc} -face.

In three dimensions, we first determine the element, P_{el}^* , that is going to be visited by the particle, on the basis of the tracking velocity and all non-eligible faces on which the particle is presently located. For an eligible element, if the normal component of the tracking velocity to each non-eligible face is

$$V_n = n_x\tilde{V}_{ix}^P + n_y\tilde{V}_{iy}^P + n_z\tilde{V}_{iz}^P \leq 0, \quad (21)$$

the particle will visit this element. Here, n_x , n_y , and n_z are the components of the outward normal unit vector on the considered face.

In the P_{el}^* -element, the plane of an eligible face is defined by

$$n_x x + n_y y + n_z z + f = 0, \quad (22)$$

and the parameter d is determined by

$$d = -\frac{n_x \xi_x^P(t_i) + n_y \xi_y^P(t_i) + n_z \xi_z^P(t_i) + f}{n_x \tilde{V}_{ix}^P + n_y \tilde{V}_{iy}^P + n_z \tilde{V}_{iz}^P}. \quad (23)$$

When $d < 0$, the particle will move away from the face plane. When $d > 0$, the particle will reach the face

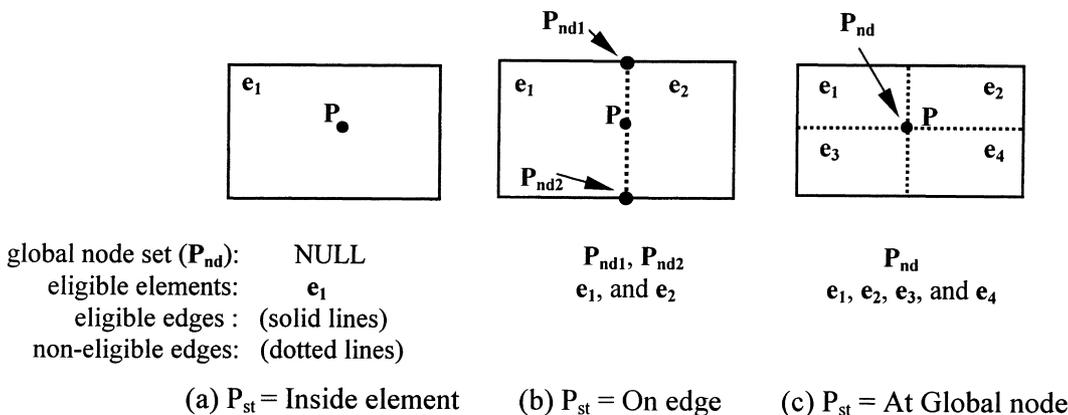


Fig. 4. Geometric demonstration for different status of particle P in two dimensions.

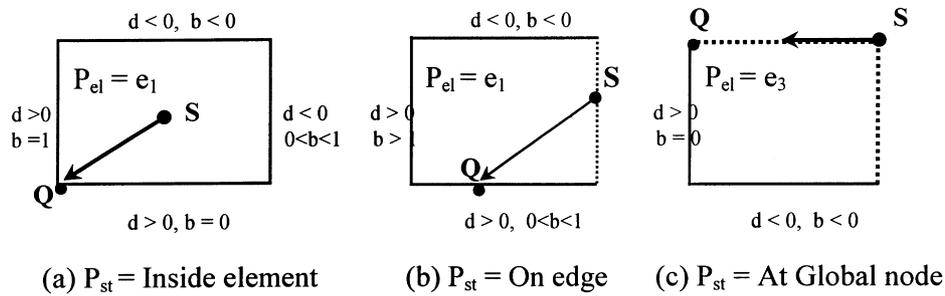


Fig. 5. Geometric demonstration for tracking a particle from point S in two dimensions. Note that the evaluation of two parameters (b and d) is made only for eligible edges.

plane if the ATTI is sufficiently large. For all the eligible faces in the P_{el}^* -element, there is a minimum positive d . It corresponds to the P_{ic} -face that the particle will cross. The minimum d is equal to the travel time, DT , of the particle from S to the exit point Q on the P_{ic} -face.

When there is no such a minimum positive parameter d , the particle is on a boundary face. It will cross the boundary face and move away from the computational domain. In this case, the particle tracking ends and its Lagrangian concentration can be determined either by boundary conditions or by concentration interpolation.

Step 3: Determine the particle's position and velocity at the trial end point.

When $\delta t_i < DT$, the particle will use up the ATTI before it exits the P_{el}^* -element. When $\delta t_i \geq DT$, the particle will reach the exit point Q before the end of the ATTI.

- The real travel time in this tracking action is

$$\Delta t_i = \min(\delta t_i, DT),$$

and the time at the end point is determined by

$$t_{i+1} = t_i + \Delta t_i = T_n + \sum_{k=1}^i \Delta t_k.$$

- The particle's position at the end point is determined by

$$\xi_j^P(t_{i+1}) = \xi_j^P(t_i) + \tilde{V}_{ij}^P \Delta t_i,$$

where $j = x, y$ in two dimensions, and $j = x, y, z$ in three dimensions.

- The particle velocity at the end point of the tracking action and at time t_{i+1} is determined by (1) or (2).

Step 4: Check the convergence of the iterative process and correct the tracking velocity.

When the average-velocity is used as the tracking velocity, an iterative process is needed to determine the particle's end point in a tracking action (shown in Fig. 3). The particle velocity at the starting point is used to predict the trial end point, and the average velocity, obtained from the particle velocity at the starting and the trial end points, is used to correct the trial end point,

until the iterative process converges. If a single-velocity is used, no iterations are needed.

3.2. Refinement and adaptation

The idea of refinement is to split the n th TTS into a number of TTIs so that the linear approximation of particle velocity produces accurate tracking in each tracking step.

For the sake of consistency with the piecewise interpolation of particle velocity, the particle tracking is performed on an element by element basis by splitting the tracking process at element boundaries. In a visited element, where the tracking accuracy cannot be met by this inter-element refinement, an in-element path refinement is needed. This path refinement is performed by splitting the travel time in such an element into a number of TTIs. The inter-element and the in-element path refinements of the particle tracking process are combined into a single set of TTIs, rather than a two-level refinement. To achieve this goal, we set a *tracking time* (ATTI) for each TTI. When a particle reaches an element boundary before it has used up the given ATTI, the inter-element refinement is performed by splitting the tracking process at the element boundary. Otherwise, the in-element path refinement is used. For the former, the travel time in a tracking step is less than the given ATTI, while they are equal for the latter. Therefore, by setting a series of ATTIs, the entire tracking process is refined into a number of tracking steps.

An ATTI is set on the basis of the local velocity field, through which a considered particle passes. It may be large when the particle velocity exhibits small variations (in magnitude and direction), and it may be small when the particle velocity undergoes significant changes. In a tracking step, when the accuracy of the particle tracking is not sufficient, a refinement of the current ATTI is performed until the above two criteria of particle velocity are met in (15). Once the tracking action has been accepted, we predict the ATTI for the next tracking step, on the basis of both the efficiency and the accuracy of

the particle tracking. The efficiency is controlled by the number of iterations required for determining the particle's end point in a tracking step. The determination of ATTIs consists of the following steps.

1. Set initial ATTI, δt_i ($i = 1$), as the travel time during which the particle passes through its first visited element, or as a predefined size of time step.
2. Perform the particle tracking action for the given ATTI, using (8).
3. Compute the particle velocity indices using (13) and (14), and check if they are within the specified ranges (e.g., using (15)).
4. If No,
 - do not accept this tracking action.
 - compute the number of sub-steps, n_t , in the tracking step

$$n_t = \max(r_i/R, \theta_i/\Theta). \quad (24)$$

- set a new ATTI as follows

$$\delta t_i = \beta \frac{\Delta t_i}{n_t}, \text{ and } \beta = \max_j \left(\frac{V_j^P(t_i) + 0.5\Delta V_{ij}^P}{V_j^P(t_i) + 0.5\Delta V_{ij}^P/n_t} \right), \quad (25)$$

where Δt_i is the travel time corresponding to ΔV_i^P . Here, we set the new ATTI such that we obtain $1/n_t$ times the displacement in the old i th TTI for the new TTI, instead of $1/n_t$ times the old TTI.

- go to Step 2.

5. If Yes,
 - accept the tracking action in the i th tracking step, and update the particle's attributes for the next step.
 - predict the ATTI, δt_{i+1} , for the next step, by

$$\delta t_{i+1} = \begin{cases} \Delta t_i/2 & \text{for } N_{it} > 10, \\ m\Delta t_i & \text{for } N_{it} < 5, r_i < R', \text{ and } \theta_i < \Theta', \\ \Delta t_i, & \text{otherwise,} \end{cases} \quad (26a)$$

and

$$\delta t_{i+1} = \min \left(\delta t_{i+1}, \Delta T_n - \sum_{k=1}^i \Delta t_k \right), \quad (26b)$$

where R' and Θ' are the limits set for the accuracy and efficiency of the particle tracking,

$$m = \min(R'/r_i, \Theta'/\theta_i),$$

and N_{it} is the number of iterations in determining the particle's end point by the average tracking velocity.

6. Track the particle step by step until the TSS is completed ($\delta t_{i+1} = 0$).

3.3. Updating the particle's attributes

After a tracking action is accepted in the adaptation step (the required accuracy is achieved), the particle's attributes need to be updated for the next tracking step.

This is based on the P_{el}^* -element and the P_{fc} -face that the particle has passed through, on the travel time Δt_i , and on the previous values of these attributes. In addition to the starting time and the particle's position and velocity determined above, the other attributes are updated as follows.

- The element that the particle is located in, P_{el} , is set to be the P_{el}^* -element.
- When $\delta t_i < DT$, the new particle status, P_{st} , may be set as being inside the element, on a face, or at an edge, by checking the new particle's position and the global nodes at all non-eligible faces of the P_{el} -element. When $\delta t_i \geq DT$, the new status of the particle may be set as being at a global node, at an edge, or on a face, by checking the particle's new position and the global nodes on the P_{fc} -face.
- Accordingly, the connected global node set, P_{nd} , are then updated, based on the new particle's status, P_{st} .
- The new ATTI for the next tracking step, δt_{i+1} , is updated in the adaptation process described above, whether the set ATTI, δt_i , is completely used up or not.

4. Numerical experiments

Let us demonstrate the accuracy and efficiency of the proposed algorithm by comparing particle tracking results in three flow situations, computed by two different methods. The average-velocity is used as the tracking velocity in each tracking step, since the single-velocity at the starting point of a tracking action is not applicable to such complicated flow fields.

The element-based particle tracking method (EPT) [4]. The average-velocity is obtained from the velocity at the starting point and at the exit point, Q (the interception of the particle path and the exit boundary face). The Newton–Raphson scheme is used for determining the particle position at the end point. In unsteady flow, the stepwise temporal average of the nodal velocity is used to approximate the temporal interpolation of tracking velocity in a single TSS [18]. When a particle cannot be tracked by the average-velocity in a visited element (this means that the iterative process is not convergent), the single-velocity is used instead. In each example, both a coarse and a fine mesh are used to demonstrate the improvement of accuracy with mesh refinement. The fine mesh is obtained by uniformly cutting the dimension of the coarse mesh by half in each coordinate direction. The nodal velocity in the fine mesh is determined by a linear interpolation of that in the coarse one.

The proposed pathline-based particle tracking algorithm (PPT). Both adaptive and non-adaptive alternatives are employed. In the latter, an identical ATTI is used in all tracking steps. Different values of the ATTI,

δt , are used to demonstrate their influence on the accuracy and efficiency of particle tracking. The average-velocity is obtained from the particle velocity at the particle’s starting and the end points within each tracking step. In unsteady flow, the particle velocity is interpolated linearly in both space and time.

Example 1. A steady circular flow [4]. The dimensionless circular flow field is defined by

$$V_x = \pi y/500 \quad \text{and} \quad V_y = -\pi x/500$$

A region of $[-3000, 3000] \times [-3000, 3000]$ is discretized by a set of rectangular elements with dimensions of 1000×1000 . The particle, initially located at $(0, 2000)$, will be at $(0, -2000)$ after a tracking period of 500. Either the travel time from $(0, 2000)$ to $(0, -2000)$, or the particle’s position at the end of this tracking period is used for demonstrating the tracking accuracy.

Table 1 and Fig. 6 show the tracking results in the different alternatives. The deviation is measured by the distance between the tracked and the exact points of a particle at the end of a travel time. In all alternatives, the particle will arrive at the exact end point $(0, -2000)$. No distortion of the tracked path is found. However, the travel time is overestimated. The EPT with the coarse mesh gives rise to a 2.4% relative error in the travel time. Or, the particle ends at point $(137, -1963)$ after the

travel time of 500. By a mesh refinement for the EPT, or a path refinement for the PPT, the errors are significantly reduced. For the EPT with the finer mesh, the error in the travel time is reduced to 0.5%.

For the PPT with equal tracking time ($\delta t = 10, 40$) in each tracking step, the tracking accuracy is improved with the reduction in the tracking time. When $\delta t = 10$, the errors both in the travel time and in the position of the end point of the particle are negligible (0.04%). For an identical number of tracking steps, the EPT with the finer mesh can produce slightly more accurate results than the PPT with equal $\delta t (= 40)$, as the latter splits the tracking process into shorter displacements in a low velocity field than in a higher one. However, the PPT with the adaptation of the tracking time, which is limited to the range of $[10, 40]$, produces identical results as the PPT with $\delta t = 10$, and a smaller number of tracking steps (13 less). Thus, the PPT with adaptation of tracking time can improve the efficiency of the particle tracking process.

Fig. 7 shows the numbers of tracking steps and iterations (for determining the particle’s end point in each tracking step) in the TSS for different ATTIs when using PPT. When $\delta t \geq 100$, the in-element path refinement does not work. Only the inter-element refinement works. The number of tracking steps (= 6) and the number of

Table 1
Comparison of the simulated travel time and the number of tracking steps in different alternatives

	EPT		PPT			Analytical
	Coarse	Fine	$\delta t = 40$	$\delta t = 10$	Adaptive	
Travel time	511.76	502.40	502.50	500.20	500.30	500
Relative error (%)	2.4	0.5	0.5	0.04	0.06	
No. of tracking steps	6	17	17	55	42	

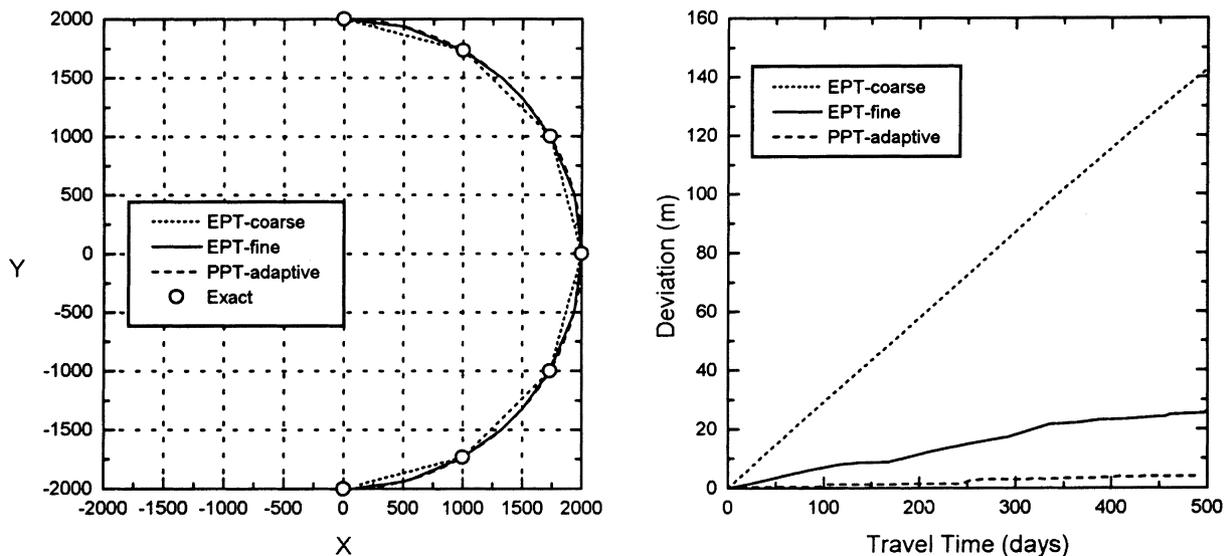


Fig. 6. Comparison of particle tracking results obtained by different methods in the circular flow.

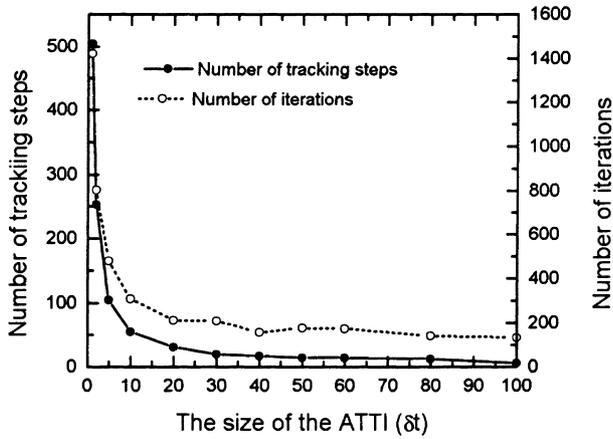


Fig. 7. Relationship between the number of tracking steps and of iterations in the TSS as functions of the ATTI size.

iterations (= 134) remain unchanged with the reduction in δt , as δt does not affect the particle tracking. When $20 \leq \delta t < 100$, the number of tracking steps is linear with the change in δt from 6 to 31. However, the number of iterations changes only slightly from 134 to 210. Thus, the reduction in δt does not significantly affect the efficiency of particle tracking, as the average number of iterations for each step decreases with the reduction in δt . When $\delta t \leq 2$, the number of tracking steps and iterations doubles when δt is reduced by half. In this case, the average of 3 iterations in each step is required. If δt is selected in the range of (5, 10), the PPT is both accurate and efficient.

Example 2. Steady flow in a two-well system. In this example, we consider an injection well, located at (0, 0), and a pumping well, located at (0, 200 m) in an infinite isotropic confined aquifer. The flow subdomain [0, 200 m]

× [0, 200 m] is considered for particle tracking. A uniform coarse mesh of rectangular elements of dimensions of 20 m × 20 m is employed (Fig. 8). The nodal velocity in the coarse mesh is obtained analytically:

$$V_x(x_i, y_i) = -\frac{K}{\phi} \frac{\partial h}{\partial x} = -\frac{K}{\phi} \sum_{j=1}^2 \left(\frac{Q_j}{2\pi T} \frac{x_i - x_{wj}}{d_{ij}^2} \right),$$

$$V_y(x_i, y_i) = -\frac{K}{\phi} \frac{\partial h}{\partial y} = -\frac{K}{\phi} \sum_{j=1}^2 \left(\frac{Q_j}{2\pi T} \frac{y_i - y_{wj}}{d_{ij}^2} \right),$$

in which

$$K = 20 \text{ m/day},$$

$$\phi = 0.2,$$

$$\frac{Q_1}{2\pi T} = -1 \text{ (for the injection well),}$$

$$\frac{Q_2}{2\pi T} = 1 \text{ (for the pumping well),}$$

$$d_{ij} = \sqrt{(x_i - x_{wj})^2 + (y_i - y_{wj})^2},$$

$$x_{w1} = 0 \text{ m}, y_{w1} = 0 \text{ m}, x_{w2} = 200 \text{ m}, y_{w2} = 0 \text{ m}.$$

Here, h is the piezometric head, K the hydraulic conductivity, ϕ the porosity, (x_i, y_i) the location of the i th node, d_{ij} the distance between the i th node and the j th well, (x_{w1}, y_{w1}) and (x_{w2}, y_{w2}) are the locations of the injection and pumping wells, respectively, T is the transmissivity of the confined aquifer, and $V_x(x_i, y_i)$ and $V_y(x_i, y_i)$ are the x and y components of the velocity at the i th node. The particle velocity at a point within an element is obtained by the bilinear spatial interpolation (1).

Three considered particles, denoted as particles I, II, and III, are initially located at points A (20 m, 0),

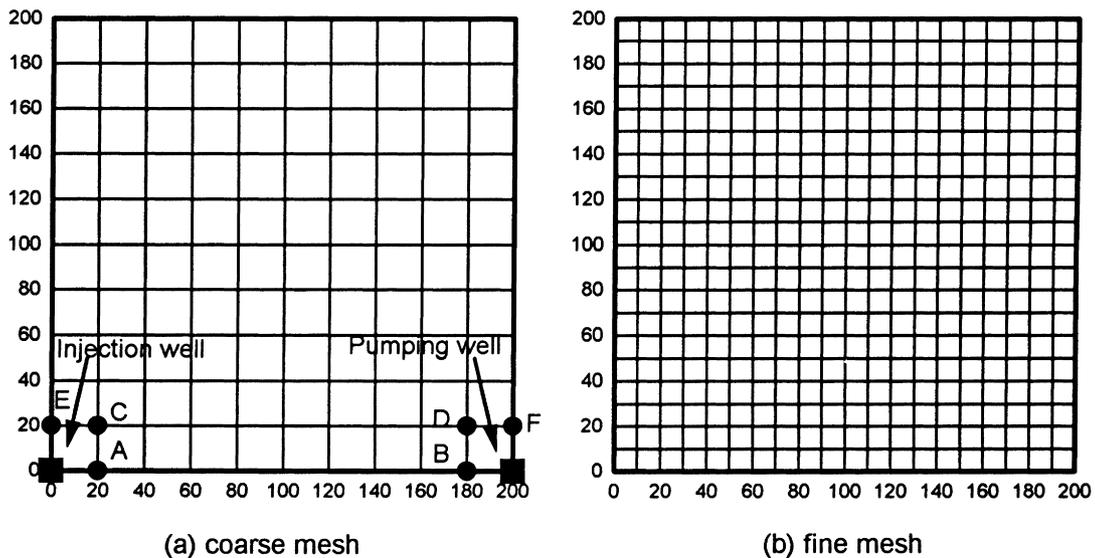
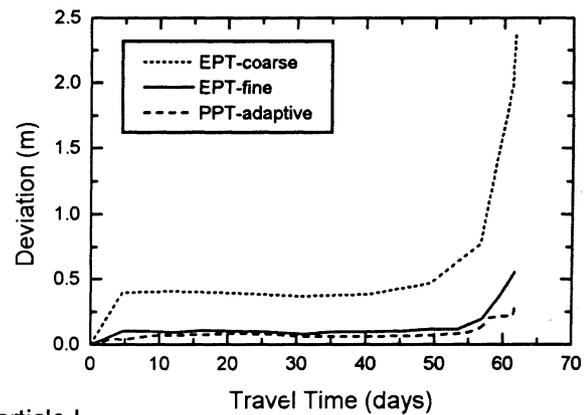
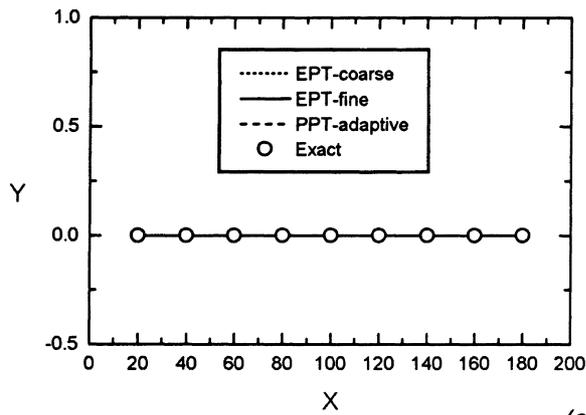


Fig. 8. Flow subdomain with coarse and fine meshes in the two-well system.

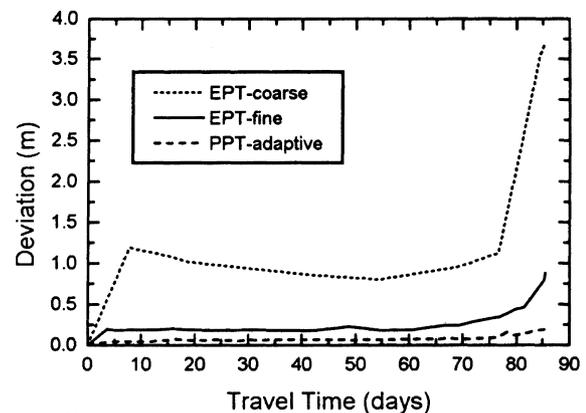
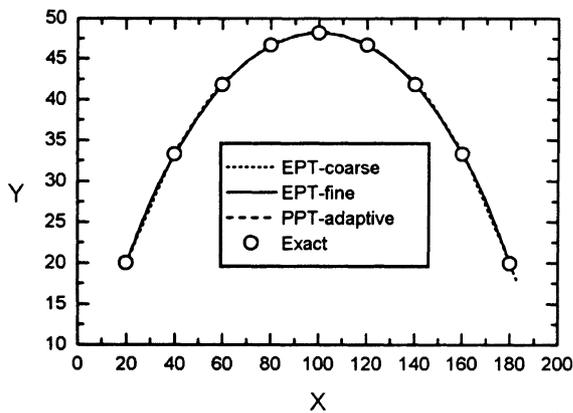
C (20 m, 20 m), and E (0, 20 m), respectively. For comparison, the exact solutions of the travel time and the displacement of these particles are obtained by the PPT with a very small tracking time, $\delta t = 0.01$ day. These exact travel times for the particles to reach their corresponding end points, B (180 m, 0), D (180 m, 20 m), and F (200 m, 20 m), are 61.63, 85.38, and 237.30 days, respectively. It should be noted that such solutions, with

respect to the assumed bilinearly interpolated velocity field within an element, may not be equal to their respective analytical ones in the real velocity field.

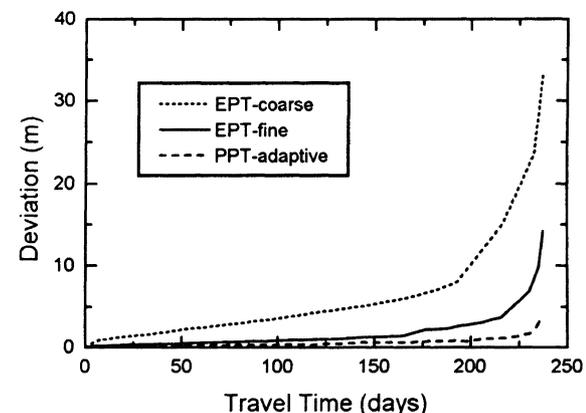
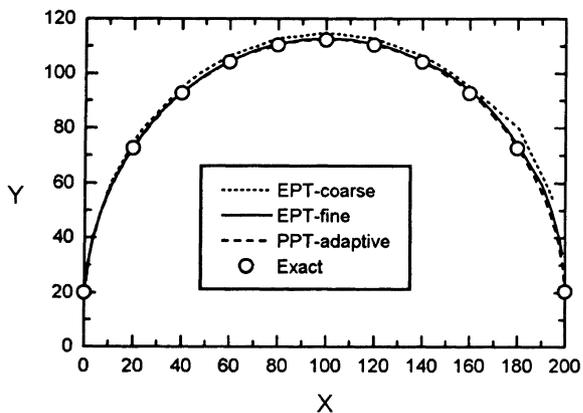
Fig. 9(a) shows that in tracking particle I along $y = 0$ from A to B, no distortion of the tracked path is found in all alternatives. However, the travel time for the given displacement AB is slightly underestimated, or the particle passes beyond the exact end point B for the given



(a) Particle I



(b) Particle II



(c) Particle III

Fig. 9. Comparison of particle tracking results obtained by different methods for steady flow in the two well system.

travel time (61.63 days). For the EPT with the coarse mesh, the particle ends at (182.4 m, 0). This error stems mainly from the tracking steps close to the injection and pumping wells, where the velocity varies significantly. By mesh refinement for the EPT or path refinement for the PPT, the error is reduced significantly. The same conclusions are reached for tracking particle II (Fig. 9(b)).

The EPT results in a large error in determining the end point of particle III for the given travel time (Fig. 9(c)). The particle ends at (195.1 m, 53.3 m) in the coarse mesh and at (198.8 m, 34.4 m) in the fine mesh. The reason is that the EPT does not track the particle by the average-velocity in four tracking steps in the coarse mesh and three steps in the fine mesh (the Newton–Raphson iterative process does not converge). Compared with the solution of the EPT with the fine mesh, the PPT with adaptation of ATTIs provides more accurate solutions at the expense of a slight increase in the number of tracking steps. It can track the particle in all steps by the average-velocity.

Fig. 10 shows that the PPT improves the accuracy of particle tracking by reducing the ATTIs in the tracking steps close to the two wells, where the velocity varies significantly in both magnitude and direction. Large ATTIs are used in regions with smooth variations in velocities.

Example 3. Unsteady flow. Here, we continue to use the same flow subdomain and the given travel times as in Example 2. Let us assume that the unsteady flow is produced by setting the x -component of the nodal velocity at the starting and end times of the TSS as 0.5 and 1.5 times its counterpart in steady flow, respectively. The y -component remains unchanged. It is assumed that the temporal velocity variation is linear in the TSS.

Fig. 11(a) shows that in the case of tracking particle I, the same results are obtained in the different alternatives for the particle’s end point as in the corresponding steady flow (Example 2). The approximation of temporally averaged velocity in the TSS produces large

errors in the middle of the tracking process. However, it does not affect the determination of the particle’s end point. In this case, the particle velocity does not change its direction during the entire tracking process. The tracking error resulting from the increase in particle velocity in the first half period of the TSS, caused by the temporal averaging, is completely compensated for by that resulting from the decrease in particle velocity in the second half. In transient flow, when the velocity field exhibits small or no variations in direction, the EPT, with this temporally averaged velocity, can result in the particle tracking being as accurate as in steady flow.

However, with both coarse and fine meshes, the EPT produces large errors in tracking particles II and III (Fig. 11(b) and (c)). A significant distortion of the tracked paths is observed. This stems mainly from the approximation of the temporal averaging of velocity in the TSS (since the travel time between the starting point and the exit point with a tracking step is not known until the exit point is determined [1,12]). In this unsteady flow, the local velocity varies significantly in its direction. Once the travel time in a tracking step is not accurately estimated, it has an adverse effect on the following tracking steps. This produces a distortion of the tracked path. Moreover, the adverse effects of this approximation are not overcome by mesh refinement.

The PPT with the adaptation of tracking time provides accurate results in tracking all three particles. The tracking time in each tracking step is automatically adapted on the basis of the complexity of the local velocity field through which the particles pass. The travel time is known (= the set ATTI) for most tracking steps, except in steps in which the particle encounters an element boundary before it completely uses up the set ATTIs. In most tracking steps, iterations are needed only for determining the end point in each TTI. In the PPT, the only approximation introduced is the replacement of the mean tracking velocity, \bar{V}_i^P , by the simple average velocity, \bar{V}_i^P . The error caused by this approximation is controlled by the two particle velocity indices. Therefore, the proposed algorithm can

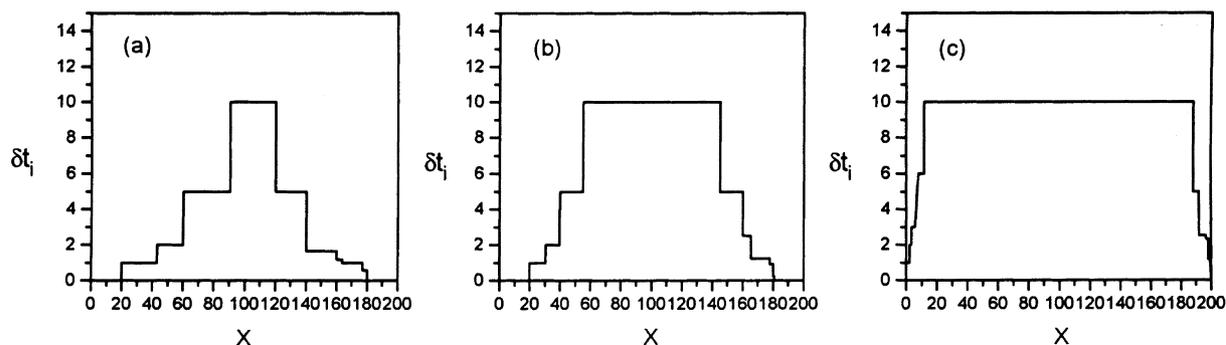
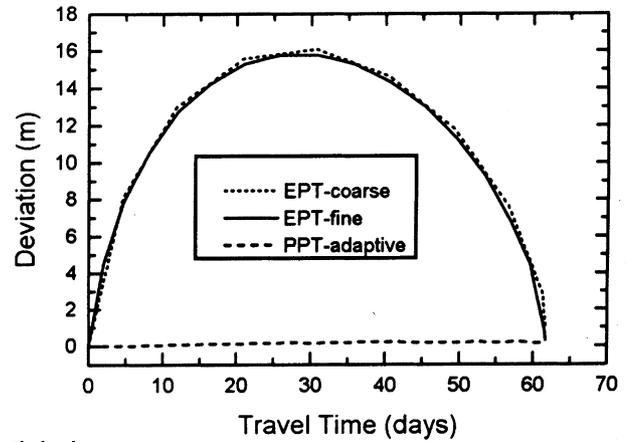
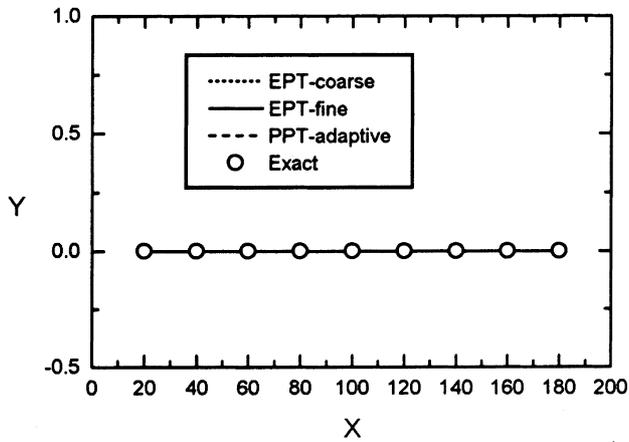
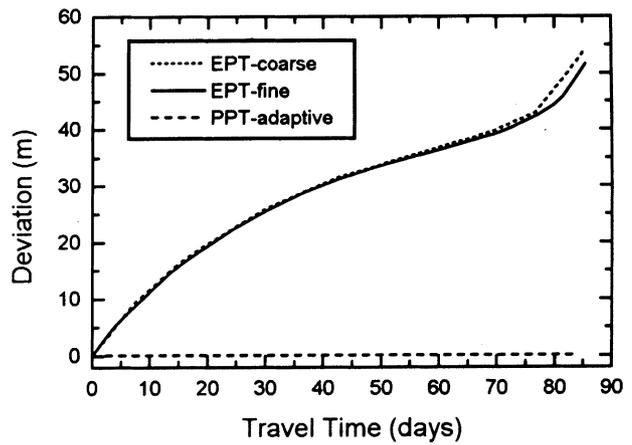
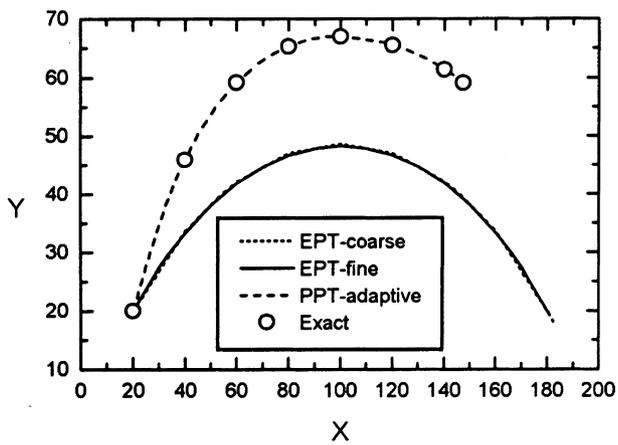


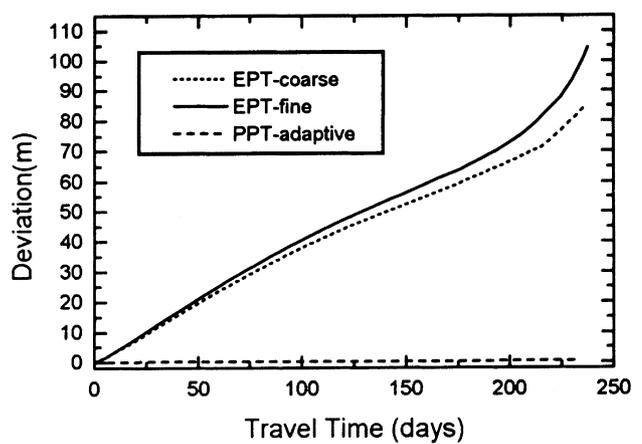
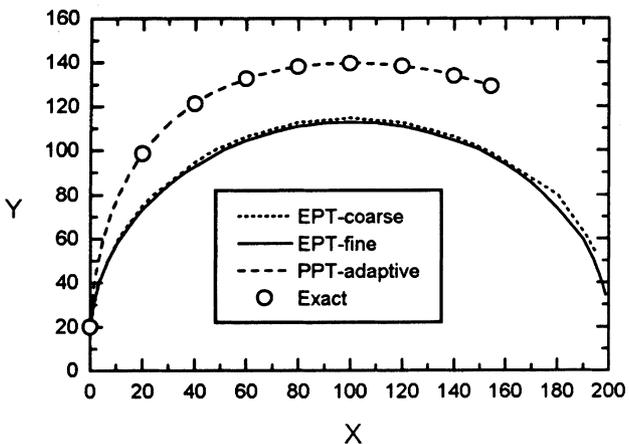
Fig. 10. Adaptation of the ATTIs (δt_i) in tracking particles I (a), II (b), and III (c).



(a) Particle I



(b) Particle II



(c) Particle III

Fig. 11. Comparison of particle tracking results obtained by different methods for unsteady flow in the two-well system.

accurately and efficiently track particles in a complicated unsteady flow. In particular, the present algorithm improves the accuracy of particle tracking in a local velocity field with significant variations in a heterogeneous flow field.

5. Conclusions

The adaptive pathline-based particle tracking algorithm is developed to improve the accuracy in the evaluation of the Lagrangian concentration in the

Eulerian–Lagrangian method for the solution of advection–dispersion problems. The main idea is to split the travel time (in a transport simulation time step) into a set of smaller travel time increments, so that the linear approximation of mean tracking velocity produces accurate tracking in a quasi uniform flow within each increment. In this way, a high accuracy is obtained, since exact tracking can be achieved for a uniform flow field. The accuracy of particle tracking is improved by refining the particle tracking process along element boundaries on an inter-element basis and by subdividing the travel time along the particle’s path into a number of travel time increments (TTIs). In this algorithm, the inter-element refinement, consistent with the piecewise interpolation of velocity, and the in-element path refinement, are combined. Tracking errors are controlled by practical criteria related to the rate of variation in particle velocity (in magnitude and direction), and they are used to ensure quasi-uniform particle velocity within a TTI. A bilinear spatial and temporal interpolation of particle velocity is used to avoid the error introduced by the stepwise temporal approximation used in most existing models. The efficiency of the particle tracking is improved by adapting the tracking time in each tracking step. This adaptation is based on two particle velocity indices and on their given criteria.

The performance of the proposed technique is compared with that of the existing element-based tracking algorithm with the help of three complicated flow situations. The results demonstrate that both the proposed algorithm, with path refinement, and the existing one, with mesh refinement, can provide accurate particle tracking results in a complicated steady flow. The proposed algorithm improves the efficiency of particle tracking by locally subdividing the tracking process in regions where the velocity varies significantly into more tracking steps than in regions with smooth variations in the velocity. However, in unsteady flow, with significant temporal variations in the velocity field in both magnitude and direction, the element-based algorithm produces large tracking errors with a significant distortion of the tracked path, while the proposed algorithm provides accurate and efficient results. Moreover, the proposed algorithm performs very well in any complex steady/unsteady flow, independent of the mesh and the size of transport simulation step.

Appendix A. Acronyms and basic definitions

Following is a list of definitions and acronyms employed in the text:

ELM	Eulerian–Lagrangian method
TSS	transport simulation time step
TTI	travel time increment in a TSS
ATTI	available travel time increment or tracking time

FDM	finite difference method
FEM	finite element method
PPT	pathline-based particle tracking algorithm
EPT	element-based particle tracking algorithm
Tracking process	particle tracking in a TSS
Tracking step	particle tracking in a TTI

References

- [1] Bear J. *Hydraulics of groundwater*. New York: McGraw-Hill, 1979.
- [2] Bear J, Verruijt A. *Modeling groundwater flow and pollution*. Dordrecht: Reidel, 1990.
- [3] Chavent G, Roberts JE. A unified physical presentation of mixed, mixed-hybrid finite element and standard finite difference approximations for the determination of velocities in waterflow problems. *Adv Water Res* 1991;14(6):329–48.
- [4] Cheng H-P, Cheng J-R, Yeh GT. A particle tracking technique for the Lagrangian-Eulerian finite element method in multi-dimensions. *Int J Numer Meth Eng* 1996;39:1115–36.
- [5] Cordes C, Kinzelbach W. Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. *Water Resour Res* 1992;28(11):2903–11.
- [6] Goode DJ. Particle velocity interpolation in block-centered finite difference groundwater flow models. *Water Resour Res* 1990;26:925–40.
- [7] Konikow LF, Bredehoeft JD. *Computer models of two-dimensional solute transport and dispersion in ground water*: Tech. of Water-Resources Invest., Ch. C2, Book 7, US Geological Survey, Boulder, Colo., 1978.
- [8] Konikow LF, Goode DJ, Hornberger GZ. A three-dimensional method-of-characteristics solute-transport model (MOC3D), US Geol. Surv. Water-Resour. Invest. Report 96-4267, 1996.
- [9] James ML, Smith GM, Wolford JC. *Applied numerical methods for digital computation*. Cambridge: Harper & Row, 1985.
- [10] Lu N. A semianalytical method of path line computation for transient finite-difference groundwater flow models. *Water Resour Res* 1994;30:2449–59.
- [11] Neuman SP. Adaptive Eulerian–Lagrangian finite element method for advection–dispersion. *Int J Numer Methods Eng* 1984;20:321–37.
- [12] Oliveira A, Baptista AM. On the role of tracking on Eulerian–Lagrangian solutions of the transport equation. *Adv Water Resour* 1998;21:539–54.
- [13] Pollock DW. Semianalytical computation of path lines for finite-difference models. *Groundwater* 1998;26:743–50.
- [14] Schafer-Perini AL, Wilson JL. Efficient and accurate front tracking for two-dimensional groundwater flow models. *Water Resour Res* 1991;27:1471–85.
- [15] Yeh GT. On the computation of Darcian velocity and mass balance in the finite element modeling of groundwater flow. *Water Resour Res* 1981;17:1529–34.
- [16] Yeh GT. A Lagrangian–Eulerian method with zoomable hidden fine mesh approach to solving advection–dispersion equations. *Water Resour Res* 1990;26:1133–44.
- [17] Yeh GT, Chang JR, Short TE. An exact peak capturing and oscillation-free scheme to solve advection–dispersion transport equations. *Water Resour Res* 1992;28:2937–51.
- [18] Yeh GT, Chang JR, Gwo JP, Lin HC, Richards DR, Martin WD. *3DFEMFAT user’s manual: a three-dimensional finite element model of density dependent flow and transport through saturated–unsaturated media*, 1993.

- [19] Zheng C. MT3D user's manual: a modular three-dimensional transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems. S.S. Papadopoulos and Associates, 1990.
- [20] Zhou Q-L, Bensabat J, Bear, J. Accurate calculation of specific discharge in heterogeneous porous media. *Water Resour Res*, to appear.